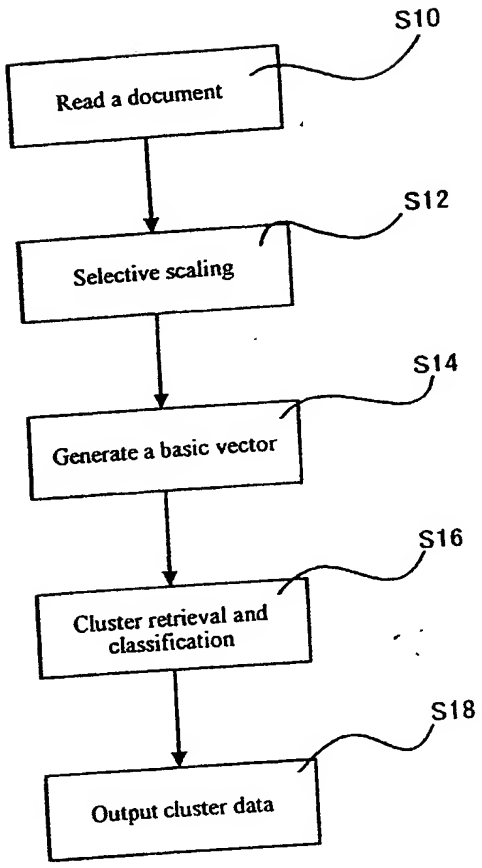

S10
Read a document
S12
Selective scaling
S14
Generate a basic vector
S16
Cluster retrieval and classification
S18
Output cluster data

Figure 1

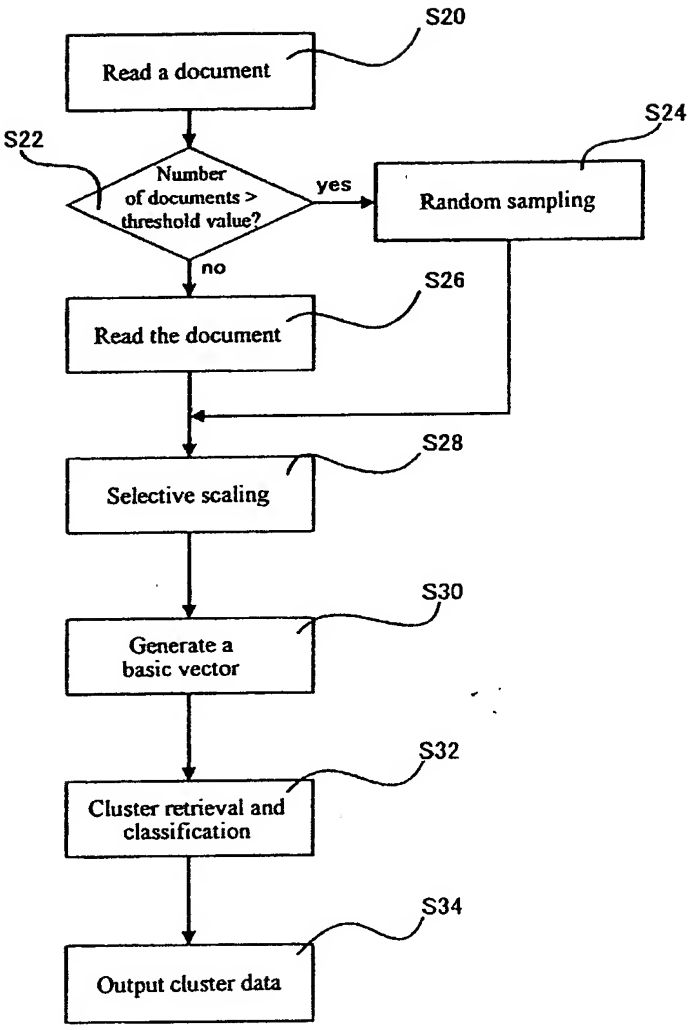

S20
Read a document
S22
Number of documents > threshold value?
yes
S24
Random sampling
no
S26
Read the document
S28
Selective scaling
S30
Generate a basic vector
S32
Cluster retrieval and classification
S34
Output cluster data

Figure 2

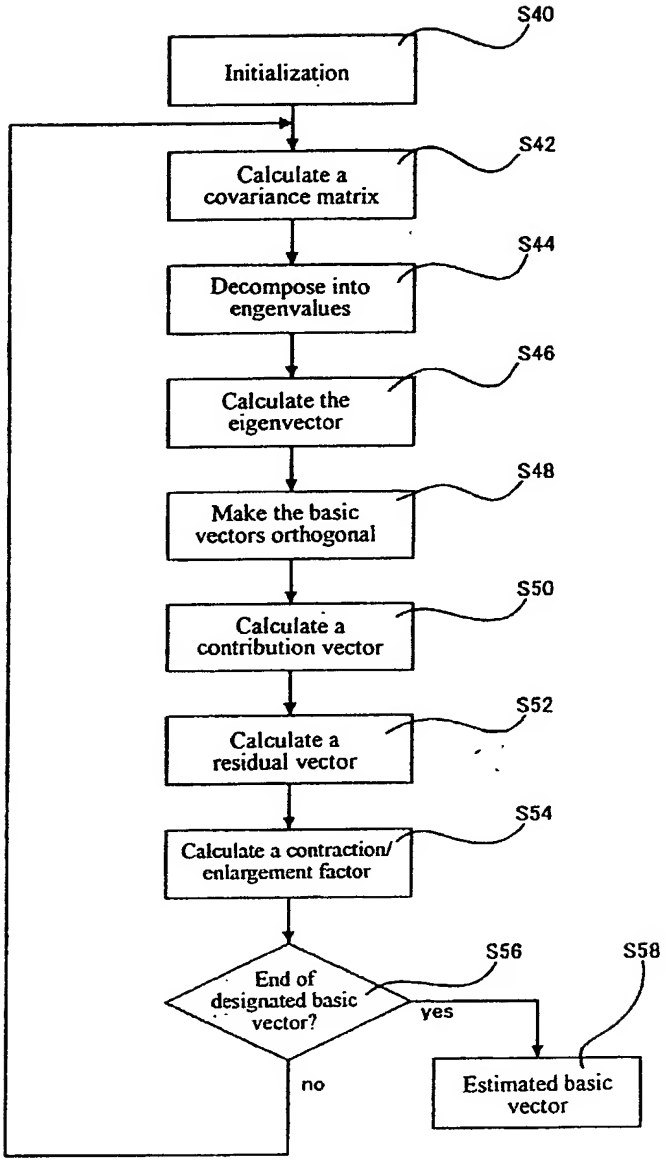

S40
Initialization
S42
Calculate a covariance matrix
S44
Decompose into engenvalues
S46
Calculate the eigenvector
S48
Make the basic vectors orthogonal
S50
Calculate a contribution vector
S52
Calculate a residual vector
S54
Calculate a contraction/ enlargement factor
S56
End of designated basic vector?
yes
no
S58
Estimated basic vector

Figure 3

```
Estimation of basic vector
Input : M'×N matrix data A'
        : Number of desired basic vectors k
        : Threshold λ
        : Scaling offset value μ
Output: k basic vectors {b_i:i=1,...,k}

(A', k, λ, μ, b){
A' : (file) pointer to M'×N matrix
R_s, R_m : M' dimensional vector
R : N dimensional vector (denotes as R_s[j] but without need for holding M'×N)
C : N×N matrix: //for holding covariance matrix
w, t : double;
first : boolean;

first = true;
for (int p = 1; p ≤ k; p++){
    if (!first){
        for (int i = 1; i ≤ M'; i++){ //step 1: selective scaling
            t = |r_i|; // obtain the length of each model vector
            if (|R_s[i]| > λ){ //inner product with basic vector is larger
                if (R_s[i] > 0.0) w = (1 - R_s[i])^(μ*t);
                else w = (1 + R_s[i])^(μ*t);
                for (int j = 1; j ≤ N; j++){
                    R_i[j] = R_i[j] × w; //scaling
                }
            }
            else continue;
        }
```

Figure 4

$\mathbf{C} = \frac{1}{M'}\sum_{i=1}^{M'} \mathbf{d}_i \mathbf{d}_i^t - \bar{\mathbf{d}}\bar{\mathbf{d}}^t$://step 2: calculate the covariance matrix

$\mathbf{b}_p = EVD(\mathbf{C})$; //step 3: eigenvector for maximum eigenvalue

$MGS(\mathbf{b}_p)$; //step 4: Modified Gram Schmidt

$output(\mathbf{b}_p)$; //output the i-th basic vector

```
for (int i = 1; i ≤ M'; i++){ //step 5: compute the contribution matrix
        R_m[i] = R_s[i] = 0.0;
        for (int j = 1; j ≤ N; j++){
                R_m[i] += R_i[j] × b_p[j];
                R_s[i] += R_i[j] × R_i[j];
        }
        if (R_s[i] == 0.0) R_s[i] = 0.0;
        else R_s[i] = R_m[i] / √R_s[i];
}
for (int i = 1; i ≤ M'; i++){ //step 6: compute the residual vector
        for (int j = 1; j ≤ N; j++){
                R_i[j] = R_i[j] − R_m[i] × b_p[j];
        }
}
if (first) first = false;
}
```

Figure 5

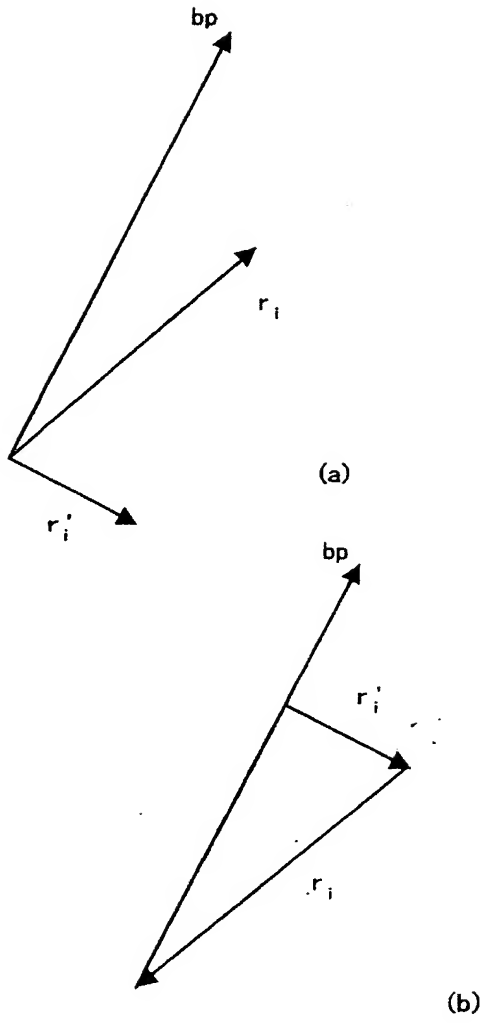


Figure 6

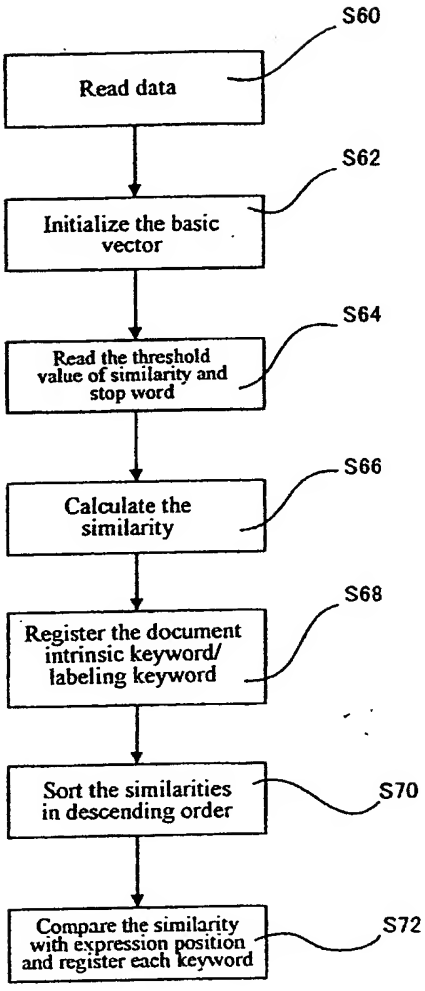

S60
Read data
S62
Initialize the basic vector
S64
Read the threshold value of similarity and stop word
S66
Calculate the similarity
S68
Register the document intrinsic keyword/ labeling keyword
S70
Sort the similarities in descending order
S72
Compare the similarity with expression position and register each keyword

Figure 7

```
1 ================================loop: r = 1↓
2 ↓
3 largest innerProduct = 0.9998399000041365↓
4 smallest innerProduct = -0.0021135015305200886↓
5 ↓
6 OK plus!--------- doc = 94636 cnt = 1↓
7 OK plus!--------- iP = 0.9998399000041365↓
8     (extrinsic) keywd = suzuki↓
9     (extrinsic) keywd = samurai↓
10    (extrinsic) keywd = japan↓
11 OK plus!--------- doc = 50840 cnt = 2↓
12 OK plus!--------- iP = 0.9743064944873534↓
13    (extrinsic) keywd = suzuki↓
14    (extrinsic) keywd = samurai↓
15    (extrinsic) keywd = sale↓
16 OK plus!--------- doc = 92853 cnt = 3↓
17 OK plus!--------- iP = 0.9372885374943962↓
18    (extrinsic) keywd = suzuki↓
19    (extrinsic) keywd = samurai↓
20    (extrinsic) keywd = sale↓
21 OK plus!--------- doc = 68088 cnt = 4↓
22 OK plus!--------- iP = 0.8239438960864272↓
23    (extrinsic) keywd = suzuki↓
24    (extrinsic) keywd = japan↓
25    (extrinsic) keywd = plan↓
26 OK plus!--------- doc = 2733 cnt = 5↓
27 OK plus!--------- iP = 0.7617615667012242↓
28    (extrinsic) keywd = suzuki↓
29    (extrinsic) keywd = samurai↓
30    (extrinsic) keywd = car↓
31    (intrinsic) keywd2 = sheriff↓
32 OK plus!--------- doc = 108212 cnt = 6↓
33 OK plus!--------- iP = 0.7123501270905321↓
34    (extrinsic) keywd = suzuki↓
35    (extrinsic) keywd = samurai↓
36    (extrinsic) keywd = car↓
37    (intrinsic) keywd2 = asher↓
38 OK plus!--------- doc = 79412 cnt = 7↓
39 OK plus!--------- iP = 0.6236521912165935↓
40    (extrinsic) keywd = suzuki↓
41    (extrinsic) keywd = maker↓
42    (extrinsic) keywd = resign↓
43    (intrinsic) keywd2 = tire↓
```

Figure 8

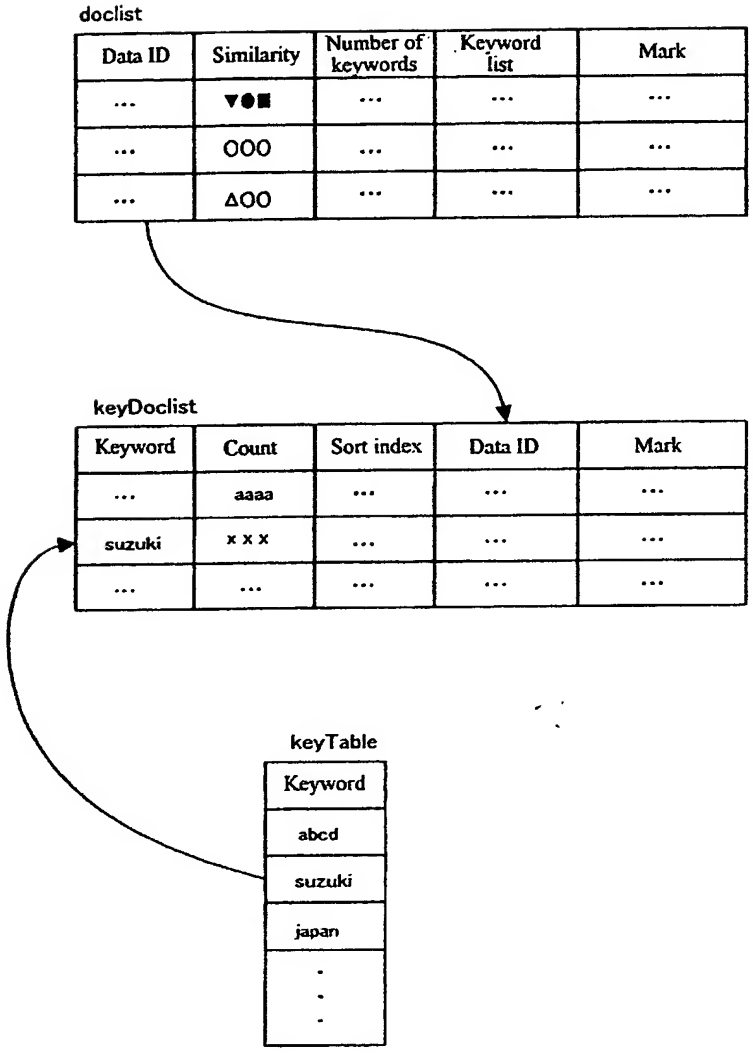

doclist
Data ID
Similarity
Number of keywords
Keyword list
Mark
▼●■
OOO
ΔOO
keyDoclist
Keyword
Count
Sort index
Data ID
Mark
aaaa
suzuki
x x x
keyTable
Keyword
abcd
suzuki
japan

Figure 9

```
Input  : k basic vectors
       : M'×N matrix data A'
       : keyword data keyword (N)
       : Number p of keywords representing cluster
       : Threshold δ for separating cluster
       : stopWord for use in cluster labeling
Output : labeled cluster set

(A', b, keyword, stopWord, p, δ, clusters){
 keyword : String[N]; // holding the keyword
 stopWord : String[]; // holding the stop word list
 b : double[k][N]; // k N-dimensional basic vectors
 minValue, maxValue : double[M'][k]; // holding the maximum and minimum of inner product
 minIndex, maxIndex : integer[M'][k]; // holding the maximum and minimum index of inner product
 model : double[N]; // holding the i-th record of A'
 keywordIndex : integer[M'][p]; // holding the index corresponding to keyword
 innerProduct : double[M']; // holding the value of inner product
 index1, index2 : integer[M']; // holding the index of data
 maxModel : double; // holding the maximum value of data
 cluster1, cluster2 : variable-length integer array; // holding the cluster data
 label1, label2 : String; // label of cluster (for output)

// step 1: initialization of various kinds of data
 cluster1 = cluster2 = null; // initialization of cluster
 label1 = label2 = null; // initialization of cluster label
 for (int r = 1; r ≤ k; r++){
```

Figure 10

```
for (int j = 1; j ≤ p; j++) // initialization of index
        keywordIndex[i][j] = -1;
maxModel = 0.0;
String line = read(1-th record of A');
while (hasMoreTokens()){ // step 2: record processing for A'
        int q = getToken(line);
        model[q]=Double(getToken());
        if (model[q]>maxValue){
                keywordIndex[i][1] = q;
                maxModel = model[q];
                j = 1;
                while (keywordIndex[i][j] ≠ -1
                        and j < p){
                        keywordIndex[i][j+1] =
                                keywordIndex[i][j];
                        j++;
                }
        }
}
double t = t1 = t2 = 0.0;
for (int j = 1; j ≤ N; j++){ // step 3: compute the similarity
        t1 =b[r][j]×model[j];
        t2+ = model[j]×model[j];
        if (t1 > maxValue[i][r]){
                maxValue[i][r] = t1;
                maxIndex[i][r] = j;
```

Figure 11

```
                    }
                    if (t1 < minValue[i][r]){
                        minValue[i][r] = t1;
                        minIndex[i][r] = j;
                    }
                    t += t1;
                }
                copyArray(index1, index2); // copying index 2 to index 1
                t = t / √t2; // normalization
                innerProduct[i] = t; // inner product with basic vector
            }
            // step 4: sorting and deciding process of cluster
            SortIndexedDescend(M', index1, innerProduct); // sorting
            // cluster process of basic vector in positive direction
            i = 1;
            while (innerProduct[i] > δ and i ≤ M'){
                addCluster(cluster1,
                        index[i], keyword, p, maxIndex[i], keyIndex[i]);
                i++;
            }
            // cluster process of basic vector in negative direction
            i = M';
            while (-innerProduct[i] > δ and i ≥ 1){
                addCluster(cluster2,
                        index[i], keyword, p, minIndex[i], keyIndex[i]);
                i--;
            }
            // step 5: cluster classification and labeling
            makeClusterLabel(r, stopWord, cluster1, cluster2, label1, label2);
            output(r, cluster1, cluster2, label1, label2);
        }
    }
```

Figure 12

	basis vector	cluster label	# of docs	percent	type
1	1+	Suzuki, samurai, car, sale, Japan	83	0.065	Noise
2	2+	Nixon, China, library, Watergate, Beijing	1342	1.051	Major
3	3+	China, Beijing, MCA, Chinese, Soviet	2662	2.084	Major
4	4+	aspen, Colorado, Chicago, condominium, tent	113	0.088	Noise
5	5+	hospital, patient, California, health, trauma	16594	12.990	Major
6	5+	Hussein, administration, oil, gulf, Jordan	978	0.766	Outlier
7	5+	campaign, candidate, measure, private, change	690	0.540	Outlier
8	6+	Iraq, Kuwait, Hussein, Iraqi, Bush	3235	2.532	Major
9	6+	Israeli, Muslim, nation, palestine, party	240	0.188	Outlier
10	6-	transplant, veronica, liver, child, organ	130	0.102	Outlier
11	7+	GM, plant, auto, Ford, car	847	0.663	Outlier
12	7-	cruise, pole, cabin, coach, Caribbean	86	0.067	Noise
13	8+	cruise, EPA, ship, apple, chemical	777	0.608	Outlier
14	9+	bird, fish, wildlife, species, endanger	719	0.563	Outlier
15	9-	school, Bush, administration, cook, child	659	0.516	Outlier
16	10+	Beijing, China, Chinese, army, troop	742	0.581	Outlier
17	10-	team, coach, school, league, UCLA	10802	8.456	Major
18	10-	Matsushita, company, Japanese, boycott, film	116	0.091	Noise
19	11+	school, Amazon, Brazil, teacher, forest	2273	1.779	Major
20	12+	team, coach, school, league, football	8307	6.503	Major
21	12-	tax, council, Bush, port, Japan	3148	2.464	Major
22	13+	duke, Louisiana, basketball, devil, campaign	381	0.298	Outlier
23	13-	tax, school, budget, council, court	167	0.131	Outlier
24	14+	school, teacher, child, education, class	3057	2.393	Major
25	14-	police, officer, arrest, cocaine, car	3167	2.479	Major
26	15-	Bush, art, school, acid, administration	1879	1.471	Major
27	15-	museum, industry, collection, Japanese, house	211	0.165	Outlier
28	16+	Bush, administration, president, congress, U.S.	2515	1.969	Major
29	16-	art, museum, Florida, music, admission	2737	2.143	Major
30	17+	team, coach, league, inning, point	9661	7.563	Major
31	17+	campaign, election, commission, Asia, councilman	766	0.600	Outlier
32	17-	school, bus, court, teacher, education	1119	0.876	Outlier
33	18+	Bush, art, museum, artist, house	2871	2.247	Major
34	18-	company, price, loan, AIDS, market	2072	1.622	Major
35	19+	council, candidate, election, campaign, school	4231	3.312	Major
36	19-	Bush, tax, art, budget, administration	1272	0.996	Outlier
37	20+	race, ascot, midget, car, park	166	0.130	Outlier
38	20-	Europe, Libya, Africa, Morocco, country	1096	0.858	Outlier
39	20-	ANC, Kadafi, ability, ally, alliance	155	0.121	Outlier

Figure 13

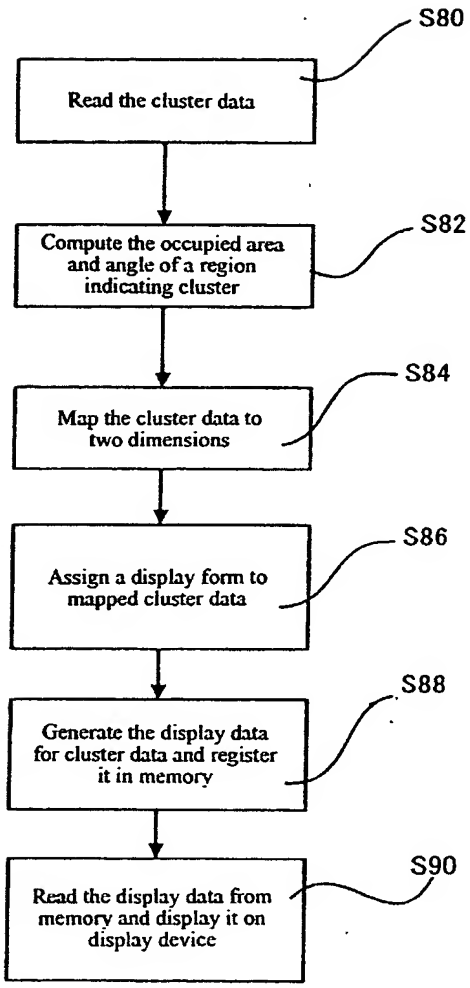

S80
Read the cluster data
S82
Compute the occupied area and angle of a region indicating cluster
S84
Map the cluster data to two dimensions
S86
Assign a display form to mapped cluster data
S88
Generate the display data for cluster data and register it in memory
S90
Read the display data from memory and display it on display device

Figure 14

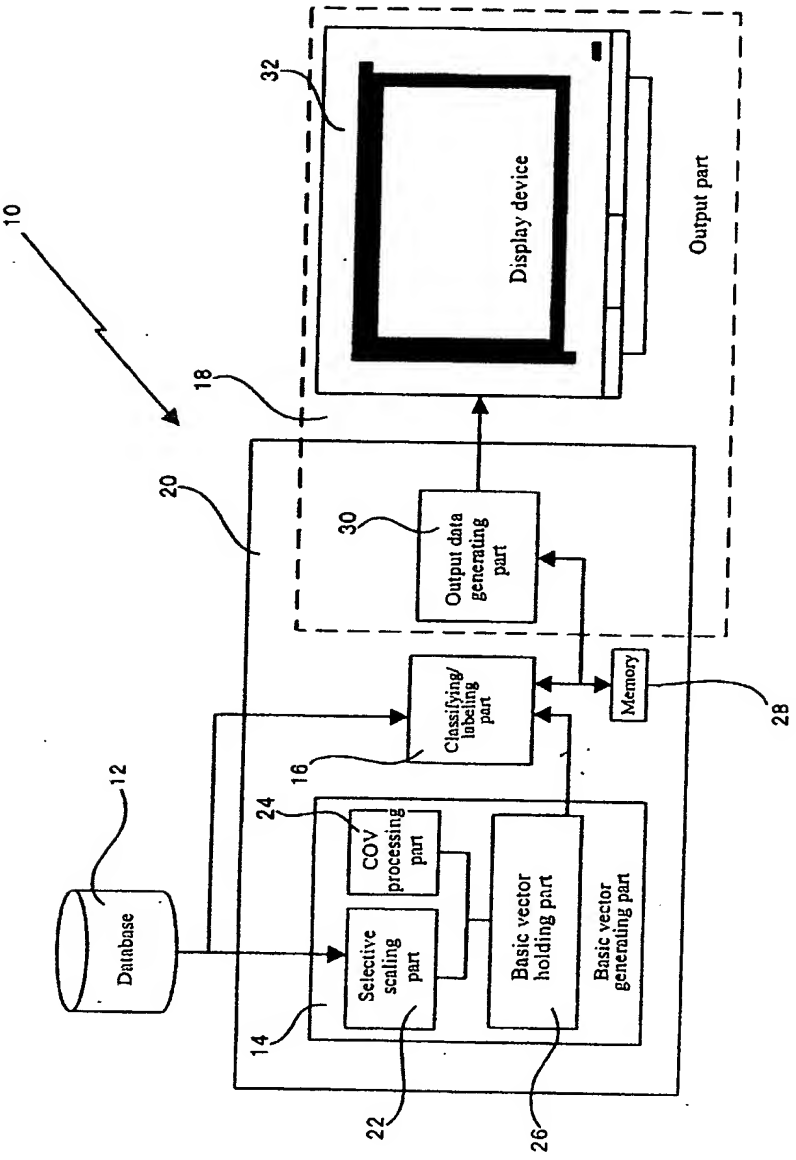

10
32
Display device
Output part
18
20
30
Output data generating part
Classifying/ labeling part
Memory
28
16
12
24
COV processing part
Basic vector holding part
Basic vector generating part
Database
Selective scaling part
14
22
26

Figure 15

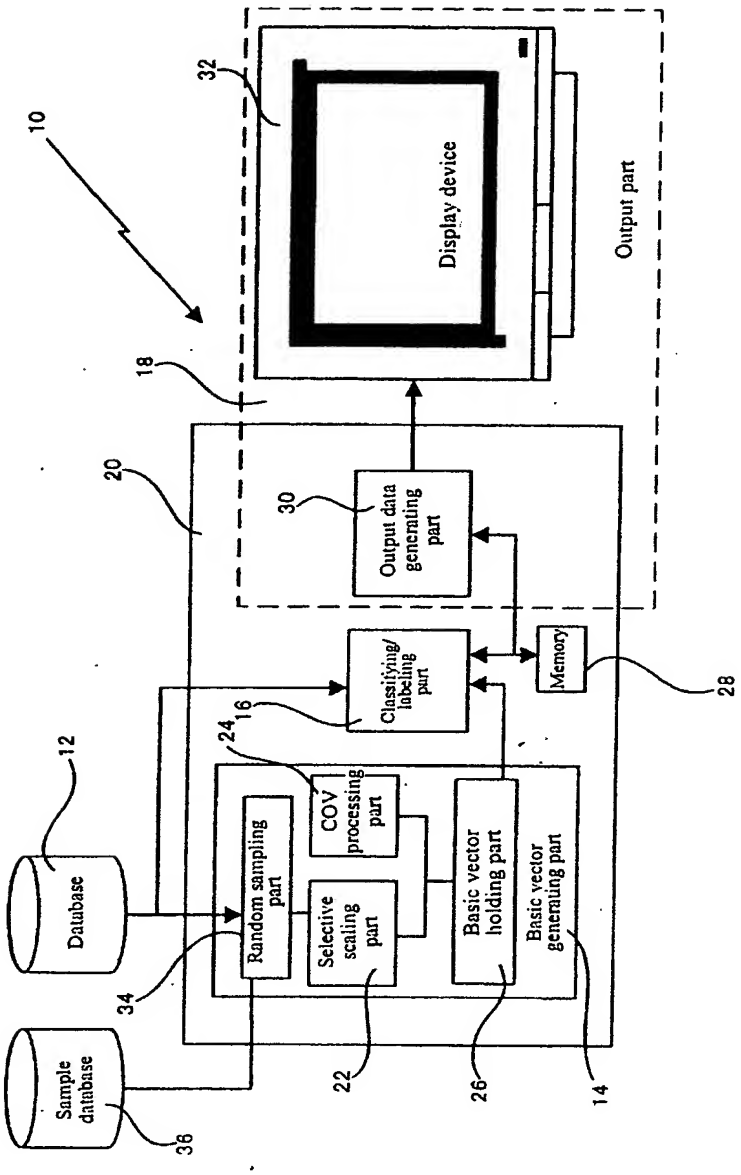

10
32
Display device
Output part
18
20
30
Output data generating part
Classifying/ labeling part
Memory
28
16
24
12
COV processing part
Random sampling part
Selective scaling part
Basic vector holding part
Basic vector generating part
Database
34
Sample database
38
22
26
14

Figure 16

Cluster name	Kind of cluster	Number of documents (%)	Set of keywords
M-1	major	4000(4%)	fruit,apple,orange,peach,banana,grape, lemon,mellon,grapefruit,strawberry
M-2	major	4000(4%)	animal,cat,dog,pig,cow,sheep, tiger,lion,elephant,monkey
M-3	major	4000(4%)	tree,maple,oak,birch,chestnut, pine,cedar,acacia,cactus,cherry
M-4	major	4000(4%)	sports,baseball,football,basketball,ski, marathon,swimming,jogging,sumo,tennis
M-5	major	4000(4%)	computer,CPU,HDD,CD-ROM,DVD, LAN,FDD,modem,memory,PCMCIA
O-1	outlier	2000 (2%)	fish,salmon,carp,tuna,caribe
O-2	outlier	2000 (2%)	vegetable,tomato,cucumber,pumpkin,lettuce
O-3	outlier	2000 (2%)	insects,butterfly,ant,beetle,dragonfly
O-4	outlier	2000 (2%)	IVY,Princeton,Cornell,Harvard,Yale
O-5	outlier	2000 (2%)	disaster,typhoon,tornado,earthquake,thunderstorm
O-6	outlier	2000 (2%)	coffee,mocha,Blue Mountain,arabica,espresso
O-7	outlier	2000 (2%)	season,spring,summer,autumn,winter
O-8	outlier	2000 (2%)	musician,Beethoven,Bach,Chopin,Mozart
O-9	outlier	2000 (2%)	mountain,Fuji,Everest,Matterhorn,Kilimanjaro
O-10	outlier	2000 (2%)	jewel,diamond,gold,pearl,ruby
O-11	outlier	2000 (2%)	entrails,heart,liver,stomach,bowel
O-12	outlier	2000 (2%)	sense,eye,ear,mouth,nose
O-13	outlier	2000 (2%)	bird,pigeon,crow,sparrow,parrot
O-14	outlier	2000 (2%)	shoes,sandal,boots,spike,highheels
O-15	outlier	2000 (2%)	river,Mississippi,Nile,Huang,Rhine
O-16	outlier	2000 (2%)	language,English,Japanese,French,Chinese
O-17	outlier	2000 (2%)	president,Kennedy,Washington,Lincoln,Roosevelt
O-18	outlier	2000 (2%)	artist,Cezanne,Van Gogh,Picasso,Renoir
O-19	outlier	2000 (2%)	color,red,white,clue,green
O-20	outlier	2000 (2%)	furniture,bed,bookshelf,cupboard,sofa
Noise	noise	40000 (below 1%)	1850 distinct keywords (absence,…,zirconium)

Figure 17

Figure 18

b_1	Positive direction				Negative direction			
	%	type	Contribution	Label	%	type	Contribution	Label
1	4.9	M	5.18	Fruit				
2	3.9	M	4.64	Animal				
3	3.2	M	2.87	Tree	4.3	M	-2.94	Sports
4	3.2	M	3.25	Computer				
5					2.9	O	-1.04	River
6	2.5	O	1.00	Fish				
7	2.2	O	0.72	Coffee	2.3	O	-0.75	Entrails
8					2.2	O	-0.97	Bird
9	2.1	O	0.79	Shoes	2.0	O	-0.68	Furniture
10	2.1	O	0.85	President	1.9	O	-0.60	Sense
11					1.8	O	-1.01	Color
12	1.8	O	0.77	Season	1.8	O	-0.72	Language
13					1.8	O	-1.00	Musician
14					1.6	O	-1.03	Mountain
15	1.5	O	0.90	Disaster				
16					1.3	O	-1.04	IVY
17					1.3	O	-1.01	Vegetable
18	1.3	O	0.96	Insects				
19					1.2	O	-1.04	Artist
20	1.3	O	0.77	Jewel				

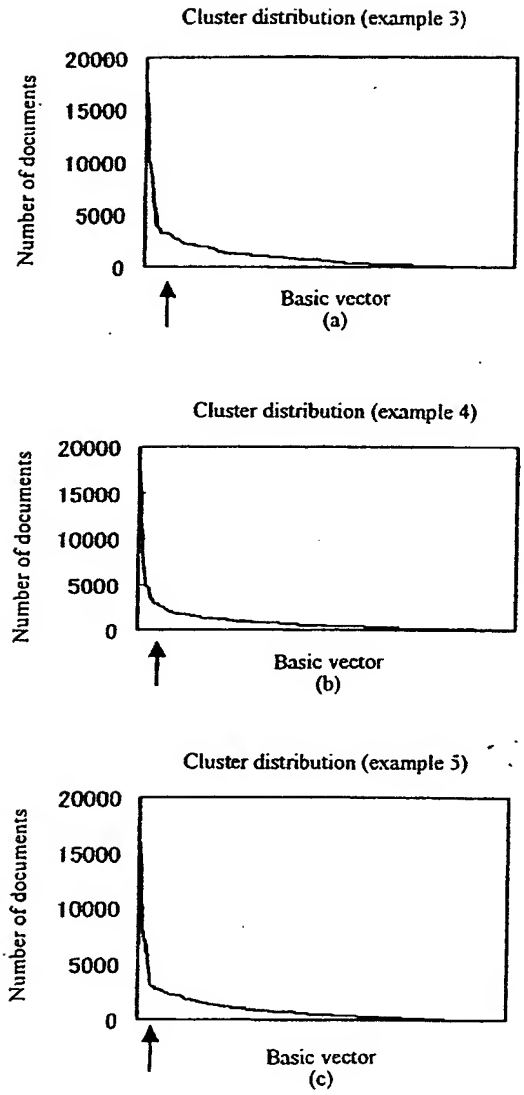

Cluster distribution (example 3)
Number of documents
20000
15000
10000
5000
0
Basic vector
(a)
Cluster distribution (example 4)
Number of documents
20000
15000
10000
5000
0
Basic vector
(b)
Cluster distribution (example 5)
Number of documents
20000
15000
10000
5000
0
Basic vector
(c)

Figure 19

FIG. 20

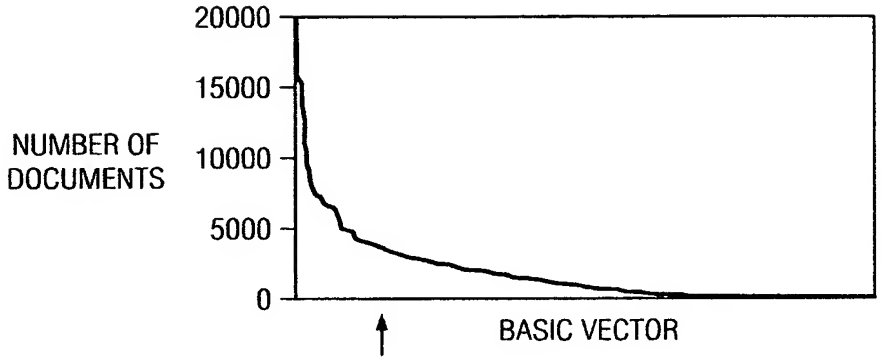
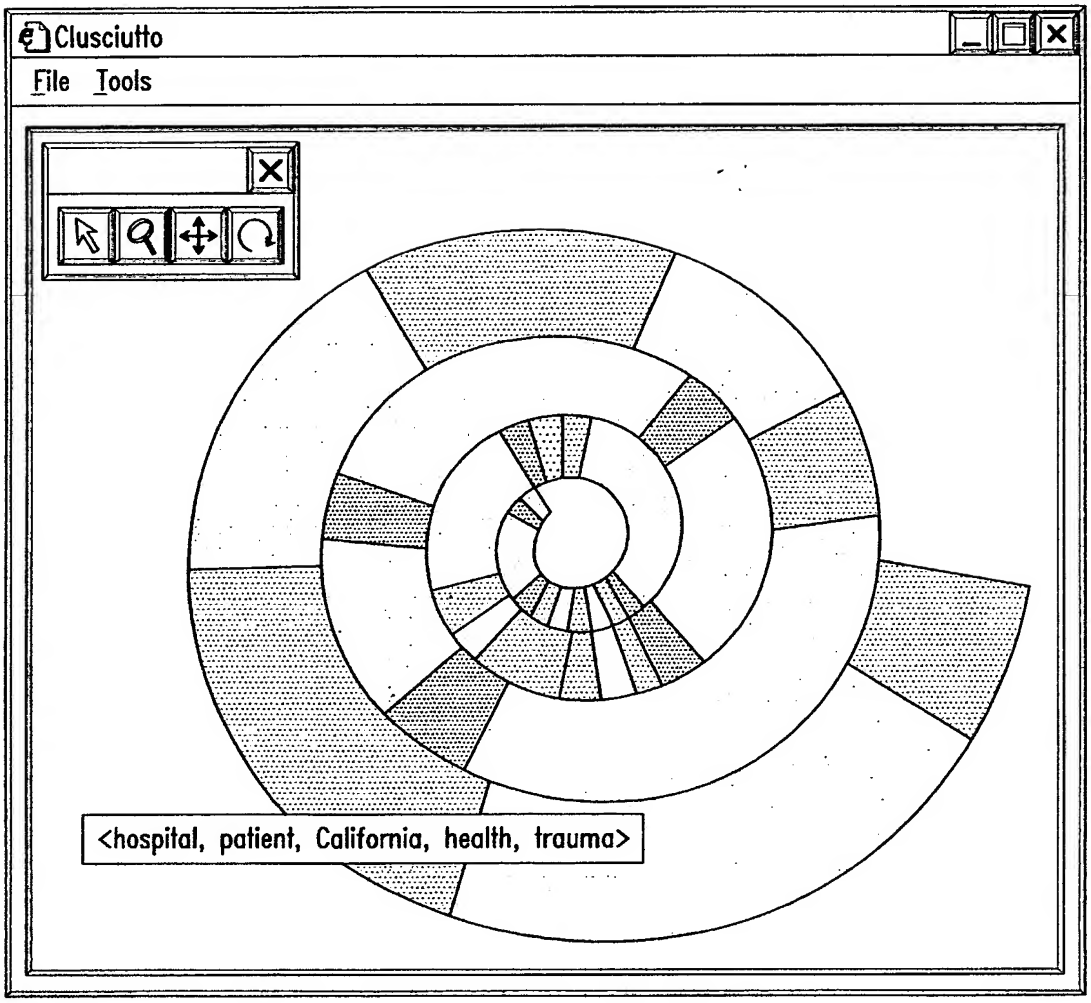

20000
15000
10000
5000
0
NUMBER OF DOCUMENTS
BASIC VECTOR

FIG. 21


Clusciutto
File Tools
<hospital, patient, California, health, trauma>

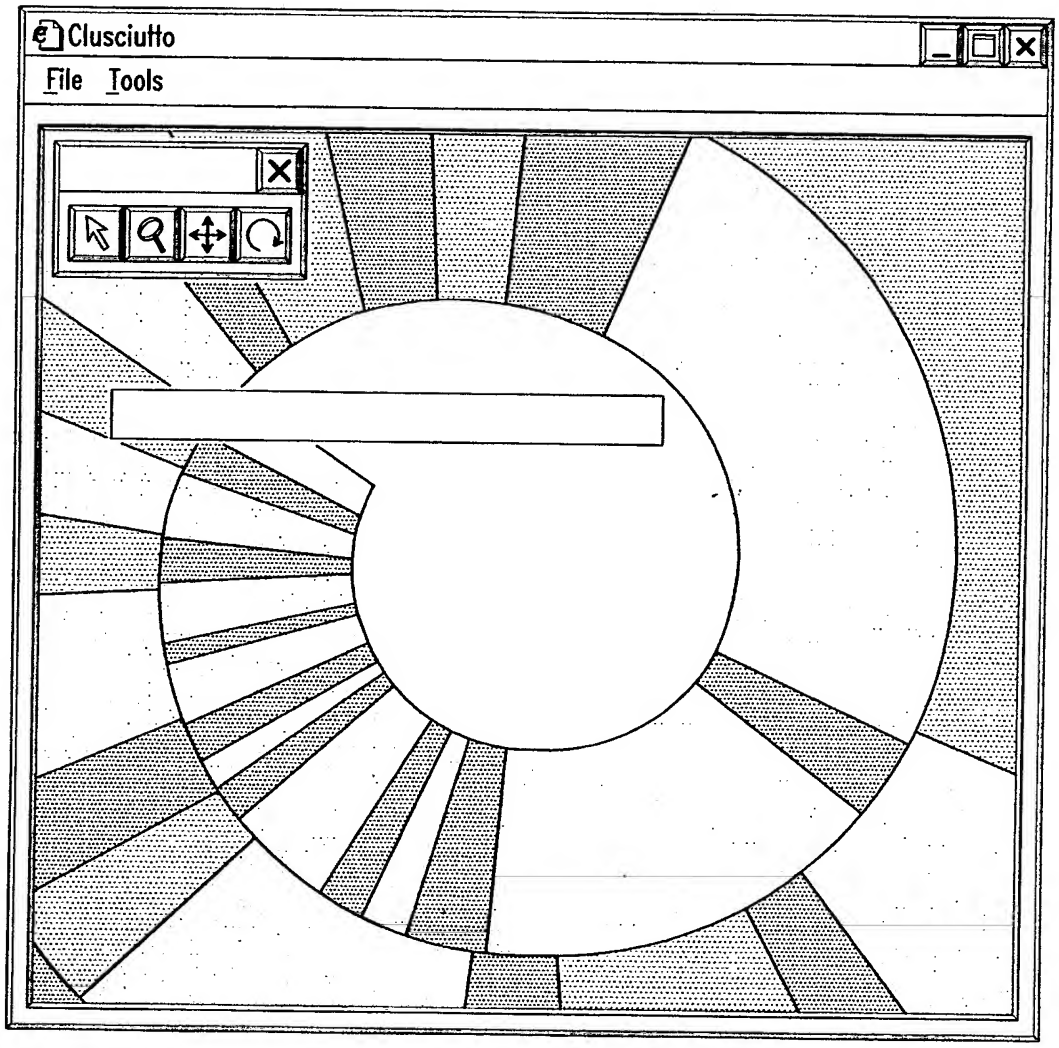


Figure 22

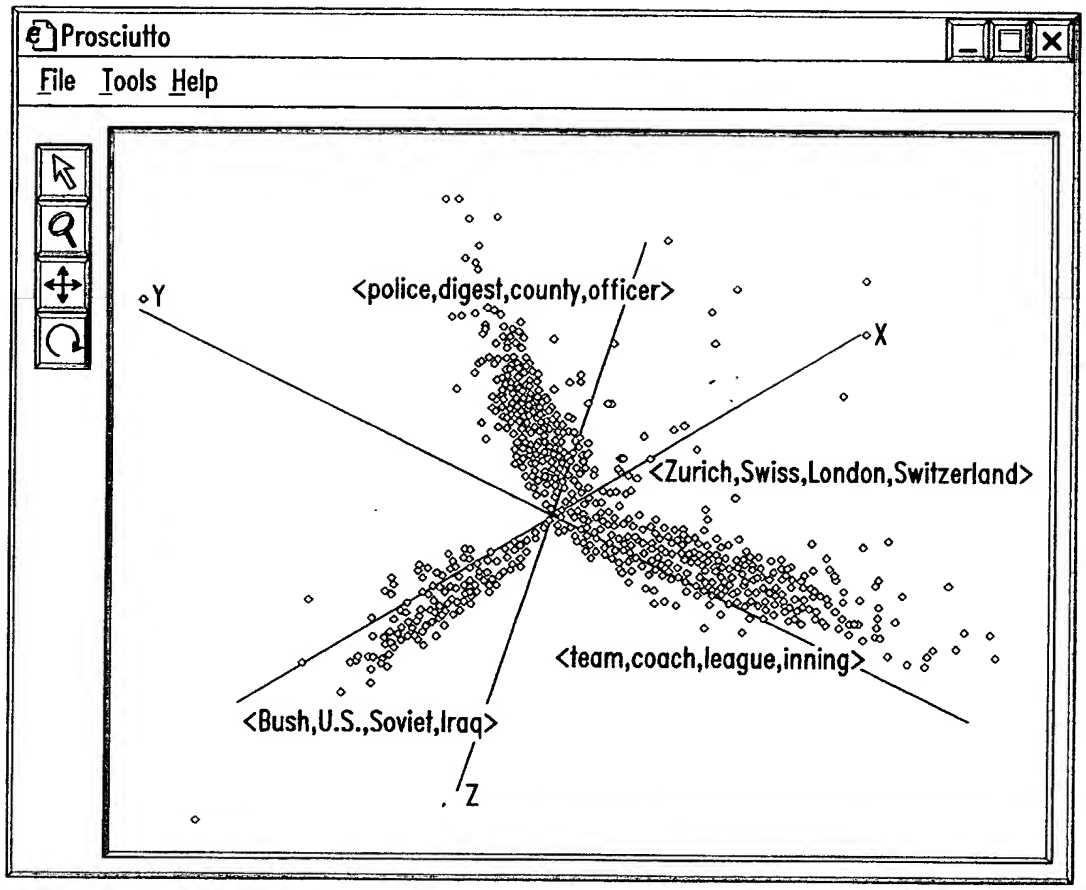

Prosciutto
File Tools Help
Y
<police,digest,county,officer>
X
<Zurich,Swiss,London,Switzerland>
<team,coach,league,inning>
<Bush,U.S.,Soviet,Iraq>
Z

Figure 23